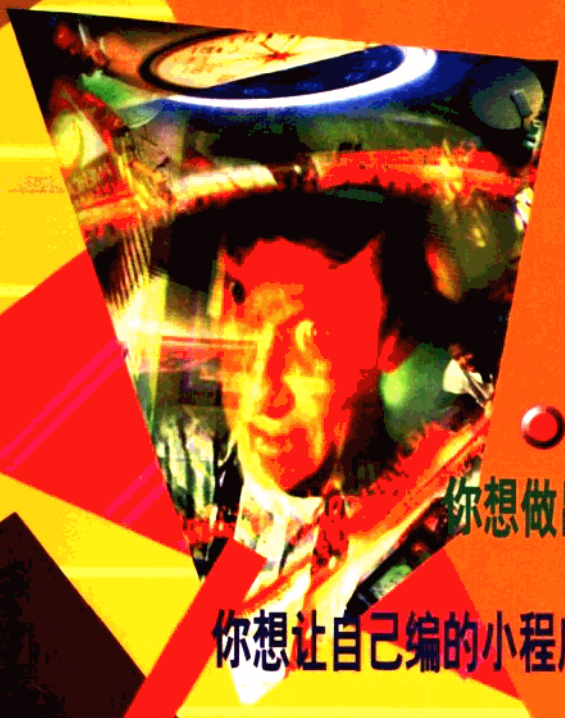
安装制作全面通

光盘＋120页软件使用手册

定价：18元



你想做出OFFICE2000的安装界面吗？

你想让自己编的小程序也可以有专业的安装界面吗？

程序员想为开发安装界面而省时吗？

全面讲解大名鼎鼎的InstallShield与WiseInstall

本产品为你提供全面解决方案！

超值附送：十多种最优秀的专业制作安装软件

前　言

《安装制作全面通》的由来还是起源于一个朋友的毕业设计，因为都是学计算机的，所以最后的毕业设计自然是编一个实用程序，最后答辩的时候虽然朋友做的程序不是最好的，但还是拿了最高的成绩，原因是它的程序很完整，用户界面很友好。是什么能使他做得这么好？这当然离不开安装制作软件的帮助，它能使自己编的软件很完整，安装界面也使人感到非常亲切，不想用的时候还可以完全卸载，看起来非常的专业化，这些繁琐的安装工作，只需几分钟就可以搞定，何乐而不为呢？

对于安装软件，我想凡是用过计算机的人都知道，尤其是现在的大多数软件都需要安装后才能运行，特别是大多数软件需要安装可以随时调用的库文件、建立程序组、图标、桌面快捷方式，有的软件还需要在注册表中添加项目。这些任务如果自己完成，会很费时间，所以现在的大公司像 Macromedia 从 Flash5、Dreamweaver4、Fireworks4 也采用了第三方的专业安装制作工具来定制自己程序的安装过程，不难看出软件开发的分工也越来越细了，使我们自己开发的小软件也可以使用专业的安装界面，会使我们的工作量减少很多，把更多的精力投入到软件开发中去。好的软件一定要有好的包装，友好的界面交互性会为你的软件增光不少。

光盘中收录了十多种最优秀的安装制作软件，其中包括很多容易上手的各种小巧安装制作软件和可以生成光盘自动播放界面的制作工具，但由于版权原因，一些大的专业软件不能在光盘中提供，我们也都附上了下载连接，包括 InstallShield 与 WiseInstall，如果读者有任何问题可以随时给我们来信，我们都会给于热情的解答。

《安装制作全面通》为读者提供了安装制作最全面的解决方案，本说明书中介绍了 12 种软件的详细使用方法，包括 InstallShield、WiseInstall、Create Install 2001、Autoplay Menu Studio Pro 等优秀的安装制作软件。使任何一个不懂编程的人，都可以轻松上手，直到变成一个专业的软件开发人员。

希望《安装制作全面通》能使您更高效地工作，迈向专业的队伍！

目　录

制　　作：宋日昕　豆　豆 姚增胜　张刘剑
排版美工：王志军　李纪平
技术咨询：　010 – 62636613
发行热线（传真）：010 – 87277636
邮购热线：010 – 87277636
邮购地址：北京市丰台区石榴园南里 18 号楼 6 门 203 室
联 系 人：周金波　赵琦
邮　　编：100075

InstallShield

图 1

一．概述

1. InstallShield 的简介

InstallShield 是 InstallShield 公司开发的一款功能强大的安装制作软件，也是目前安装软体中最为流行，最受广大用户喜爱的产品。它具有操作简单、完全可视化的界面、方便的调试功能、支持多平台语言的特点。而在 Windows 平台上，Office、VC + +、VB，WPS2000 等几乎所有软件都有着令人炫目的安装界面，方便的组件选择，自动的程序卸载功能，这些都应归功于 InstallShield 公司的杰作。由于 InstallShield 具有如此完美的特性，微软的 VC6 捆绑了 InstallShield，以便 VC6 制作的应用软件，在优越的性能之外，更不能缺少美丽的外观。而今，随着日月的推移，InstallShield 更是羽翼丰满，又推出 InstallShield 6.0，功能更为强大，操作更为简便。对于一个软件的制作者，包装上一个漂亮的安装程序，实在是非常必要。本文主要介绍 InstallShield 制作安装软件的使用方法，希望对感兴趣的朋友有所帮助。公司其它的软件还有：

a. PackageForTheWeb
b. InstallFromTheWeb
c. RegisterOntoTheWeb
d. InstallShield Express
e . InstallShield Java Edition
f. DemoShield

2. 公司新闻

4 月初，IBM 公司宣布，它将与 InstallShield 软件公司合作为客户提供一致的、便于使用的安装解决方案。该解决方案提供了跨平台的标准化安装流程，网络管理人员只要掌握了"Windows 式"安装程序，即可实现软件产品的多平台安装，从而大大提高工作效率。

借助这项联合研究计划，IBM 软件及运行于 AIX、AS/400、Linux、OS/2、Project Mon-

terey 和 Solaris 平台的 ISV(独立软件开发商）应用软件的最终用户将可获得与 InstallShield 当前提供给 Windows 用户一致的、便于使用的安装流程。

根据协议，来自 InstallShield 和 IBM 的开发工作组将合作开发下一版 InstallShield Java 软件。这一专为从事多平台软件开发的软件开发人员设计的解决方案，将使管理人员和最终用户利用当前基于 Windows 的应用软件安装中所用的对话框进行跨平台安装。

IBM 业务合作伙伴以及为 IBM 平台开发软件的 ISV 也会从这项协议中受益。其开发人员可以获得一个单一的、高度灵活的开发环境，使跨产品集成变得更加方便，从而有效地提高安装程序代码在多种操作系统中的可复用性。

3. InstallShield 的安装

安装软件对于高手来说当然是非常简单，但是为了更多的初学者的需求，也就不嫌罗嗦介绍一下：

a. **密码提示画面**：如图 2

请执行所下载的 InstallShield 的安装文件！Ipse622. Exe，然后会有欢迎画面，接下来就要在 Password 中输入密码。

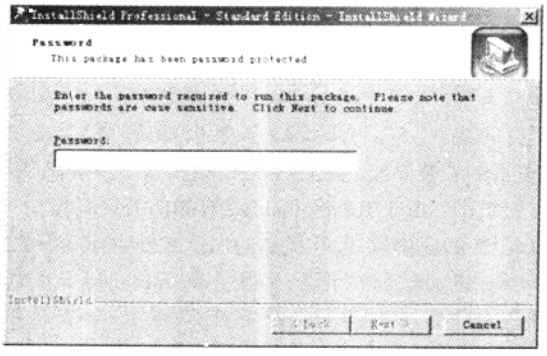


图 2

b. **安装文件解压路径画面**：如图 3

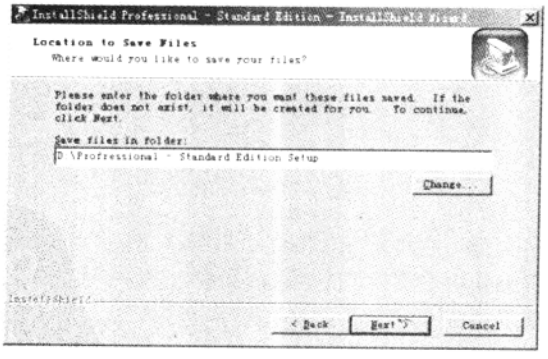


图 3

在 save files in folder 中输入文件解压路径，或点击 chang 按钮进行选择。完成后，按 Next 继续。

c. **解压状态画面**：如图 4

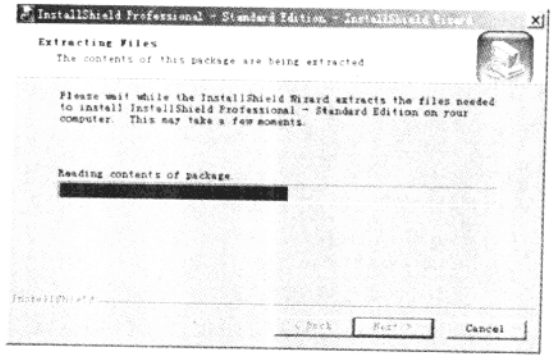


图 4

软件将安装所需文件到解压指定路径下，并显示进程。

d. **欢迎画面**：如图 5

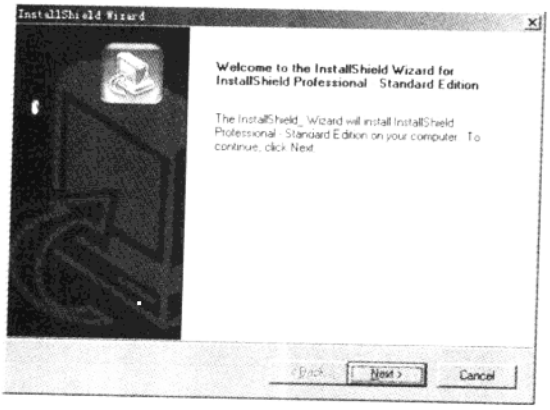


图 5

提示：欢迎安装该软件，点击 Next 继续。

e. **授权同意画面**：如图 6

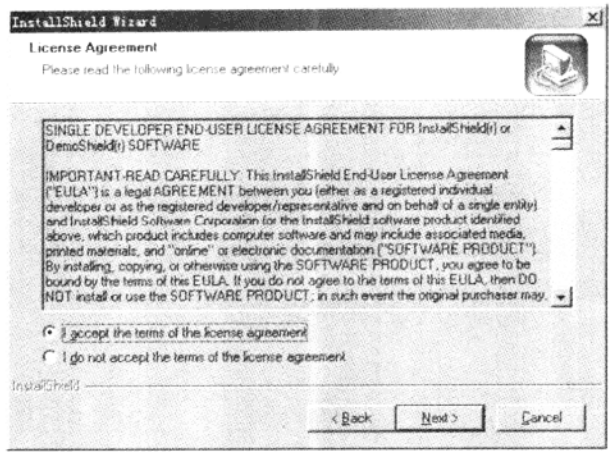


图 6

在此画面中，如果你不同意 InstallShield 公司的授权同意书的话，选中 I don't accept 离开，同意，选中 I accept，单击 Next 继续。

f. **使用者资料画面：**如图 7

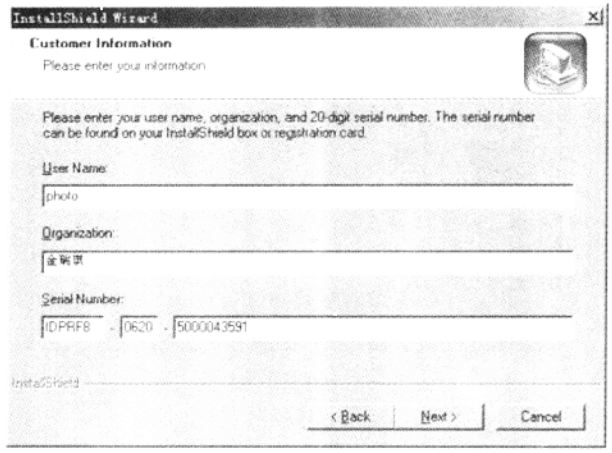


图 7

在本画面中，你必须输入使用者名称及公司名称，而且最重要的是你也要输入此软件的安装序号，如果安装序号没有输入或是不正确的话，那都无法进行下一步安装，单击 Next 继续。

g. **选择目的地路径画面：**如图 8

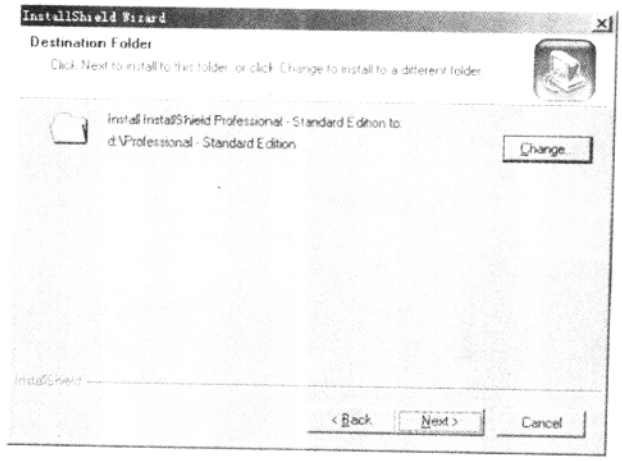


图 8

此画面会显示出本软件将会被安装到的目的地路径，点击 Change 按钮开启对话框，选择自己想要安装的目的地路径，单击 Next 继续。

h. **选择安装类型画面**：如图 9

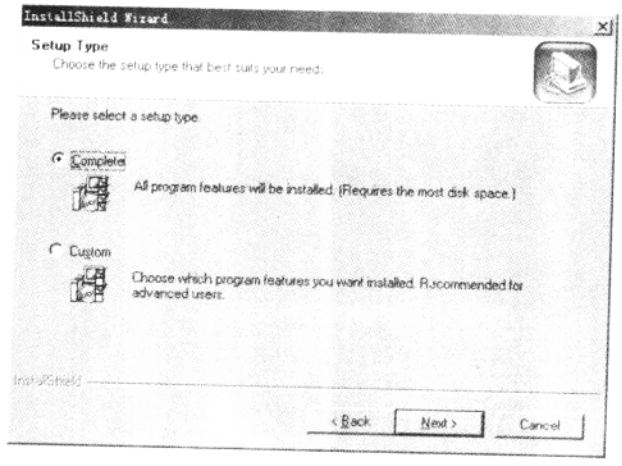


图 9

Complete: 完全安装，即安装所用文件；

Custom: 自定义安装，可以选取组件；

选中 Complete，单击 Next 继续。

i. **准备开始安装画面**：如图 10

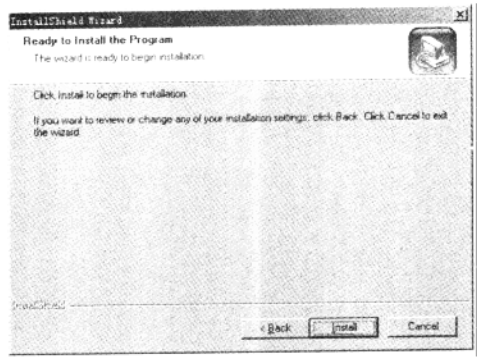


图 10

如果一切都准备就绪了，点击 Install 按钮开始进行安装。

j. 安装进程画面：如图 11

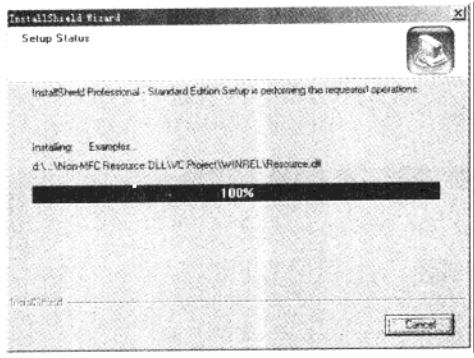


图 11

k. 安装结束画面：如图 12

图 12

单击 Finish，继续。

1. **软件注册画面**：如图 13

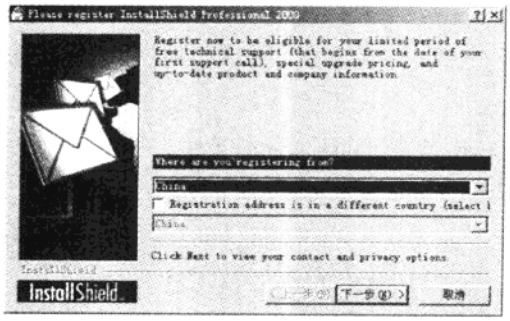


图 13

单击下一步进行注册，单击取消，完成安装。

二．InstallShield 操作环境，如图 14

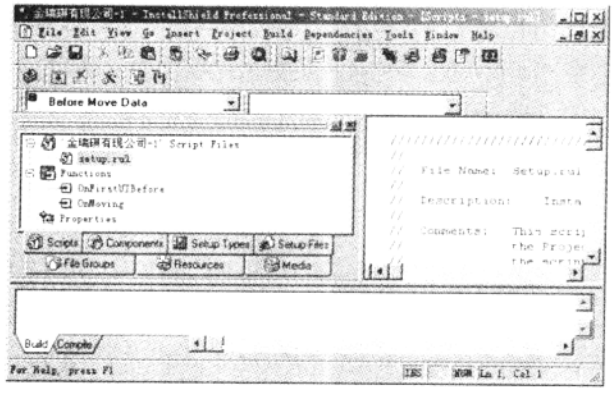


图 14

1. **菜单栏**，如图 15

包括文件(File)、编辑(Edit)、查看(View)、跳转(Go)、插入 (Insert)、项目 (Project) 、创建(Build) 、调试(Dependencies)、工具(Tools)、窗口(windows)和帮助(Help)七个菜单

图 15

2. **工具栏**，如图 16

可以通过点击图标实现功能的操作，如打开，存储项目，插入函数，打开资源管理器，创建安装文件，脚本的调试等。

图 16

3. **状态窗口**，如图 17

共分为七大模块：Scripts(脚本)，Components(组件)，SetupTypes(安装类型)，Setup Files(安装文件)，File Groups(文件组)，Resources(资源)，Media(安装介质)。基本上创建文件的工作都可以在这里完成。

图 17

4. **脚本编译窗口**，如图 18

进行脚本程序编译，非常简单，划分也很明确，出现的子窗口与操作的状态对应。

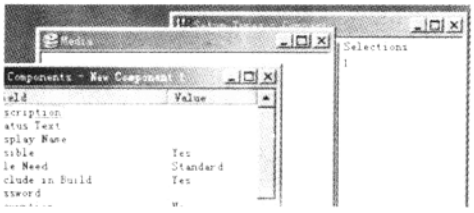


图 18

三．脚本语言

下面主要介绍脚本语言中的数据，函数，流程控制。

A. 数据

1. 数组

能将安装脚本数据定义为数组类型，例如 NUMBER nArray(10)；如果不定义数组长度，如 NUMBER nArray()；数组长度将被默认为零。

使用 Resize 重新定义长度，格式 Resize(Array, nNewsize) Array 数组变量名称，nNewsize 为新的长度，返回值是定义了新长度的数组变量。

使用 Sizeof 命令定义长度。

使用下列语法设定数组变量：

<array variable name> (<array index>) = <value> ;

例如:nArray(5) = 17;或 nArray(0) = 1;(编号从零开始)

2. 常量

常量是已被赋值的数据项,InstallShield 支持两种类型的常量。

a. Predefined constants 事先规定的常量,类似 TURE、RESET,是安装脚步的一部分,这些常量在函数参数中调用,值返回到内置函数中。不能在脚本中被重新定义,如果试图定义 Predefined constants,将导致编译出错。

b. User - defined constants 用户定义常量,这类常量是根据特别脚本的需要,程序编制员自行定义的,能被重新定义。

3. 结构数据

结构数据用来指定数据集中相关变量和命名成员。在大多数编译语言中,结构数据用于命名数据记录。数据结构在安装脚本中的使用与 C 语言中相似。它能包括不同类型数据成员,那些成员内部的结构数据能直接使用成员命令。例如

```
typedef EMPLOYEE
begin
   STRING szName[50];
   STRING szDepartment[50];
   NUMBER nExtension;
end;
```

数据结构名为 EMPLOYEE,包括三个变量。

4. 数据类型,数据类型包括以下几种:

a. BOOL 布尔型,是一种逻辑类型,只包括真或假,真为 1,假为 0。

b. CHAR 字符型,一字节长的(8bit)的字符

c. HWND 窗口句柄用来存放窗口句柄

d. INT 整形,与 NUMBER 等价,两字节长的整数

e. LIST 列表型指向 InstallShield 列表,用于 ListCreate 和 ListDestroy 两个函数

f. LONG 长整形

g. LPSTR 扩展指针

h. NUMBER 数字型,占用 4 个字节的整形,范围在 - 2147483648 ~ + 2147483647 之间。注意在 InstallShield 中所有 numeric 数据类型同 Nember 变量类型一样。

i. OBJECT

j. POINTER 指针型

k. SHORT 短整形

l. STRING 字符数组型

m. VARIANT 所用类型,可以用它定义不确定的类型。

5. 特殊字元

a. \n 回车

b. \? 在字符串中插入单引号

c. \t 插入空格

d. \ooo 转换为 8 进制数

e. \xhh 转换为 16 进制数

f. \\插入反斜杠

6. **操作符**

一般的与C语言相同操作符，在这里不做详解，以下主要介绍比较特殊的操作符。

(1) +，-，*，/

以上四个操作符与C语言中意义和用法都相同。

(2) &&

与操作，与C语言中用法相同，例：x1 && x2

(3) ||

或操作，与C语言中用法相同，例：x1 || x2

(4) !

非操作，与C语言中用法相同，例：!x1

(5) *

指针操作，类似C语言中的*

(6) &，||，^，~，<<，>>

分别为与，或，按异或，取反，左移和右移，其意义和用法都与C语言中基本相同。

(7) .

该操作符用于结构，用来得到结构的子项，与Delphi的用法类似，例如：

```
typedef  SETTINGSREC
begin
BOOL bSwitchOn;
STRING szMssg[255];
INT nVal;

end;
SETTINGSREC settings;

program
settings. bSwitchOn  =  FALSE;
settings. szMssg  =  "Off";
settings. nVal  =  0;
```

(8) =

既可作为赋值号，同时也做等于符，例如：

```
str1 = "String";
if str1 = "String" then
endif;
```

(9) &

取地址符，与C语言用法类似。

(10) <，>，=，<=，>=，!=

分别表示小于，大于，等于，小于等于，大于等于，不等于

(11) +，^，%

用于字符串的操作。

(12) ->　结构指针，与C语言中用法类似。

(13) @

用于得到Resource窗口中定义的字符串，例：

```
szReferenceFile = svDir ^ @ PRODUCT_KEY;
```

7. 全局与局部变量

全局变量可以在整个编译脚本被调用，局部变量只能在的定义函数体中调用

```
STRING szVal;              //定义全局变量
prototype AFunction(STRING);
szVal = "YES";
AFunction(szVal);
MessageBox(szVal, INFORMATION);

function AFunction(szVal)
STRING sz                  //定义局部变量
Begin
    sz = "yes";
    szVal = "NO";
end;
```

8. 变数命名规则

在变量加...	变量类型	说明
b	Boolean(BOOL)	布尔型常数、literal或布尔型变量
bv	Boolean(BOOL)	只限于布尔型变量
c	Character(CHAR)	字符常数、literal或布尔型变量
const	Constant	常数或literal
h	Handle(HWND)	Handle变量
i	Integer(INT)	整数常数、literal或整型变量
l	Long Integer(LONG)	长整型常数、literal或长整型变量

lv	Long Integer(LONG)	只限于长整型变量
list	List(LIST)	List 变量
n	Number(NUMBER)	数字常数、literal 或数字变量
nv	Number(NUMBER)	只限於数字变量
p	Pointer(POINTER)	指标变量
pstruct	Pointer to a uetined stucture type	Not Used
s	Short Integer(SHORT)	短整型常量、literal 或短整型变量
sz	String(STRING)	字符串常量、literal 或字符串变量
sv	String(STRING)	只限于字符串变量
struct	Defined structure type	Not Used

9. 指针

指针是一个地址变量，声明格式：POINTER pPointerName

例如：

```
typedef RECT          //声明一个结构类型
begin
    SHORT sX;
    SHORT sY;
end;
RECT Rectangle;       //声明一个此类型的变量
RECT POINTER pRect;   //声明一个此类型的指针
```

10. 变量

声明变量的格式：

```
data type  variable name[variable name[. . ]];
```

注意变数名称不得超过 32 字符

例如：

```
BOOL bValidEntry; // 声明一个布尔型变量
LONG lPopulation; // 声明一个长整型变量
STRING szUserName[128]; //声明一个规定长度的数组变量
NUMBER nFileSize, nDirSize, nDiskSpace; // 声明三个数字型变量
```

B. 函 数

InstallShield 的函数使用前同样需要声明，函数的参数传递方式十分类似 C 语言，例如下面的函数声明：

```
prototype HandleMoveDataError( NUMBER );
```

该声明中函数名为 HandleMoveDataError，传递一个 NUMBER 类型的参数。调用该函

数时也基本与 C 语言中相同 。

a. **函数体的标准格式为 :**

```
Function functionname(nResult)// 函数变量声明区
begin
... ...// 程序区
end;
```

通常的函数返回一个数字型的数 。

b. **框架程序的基本结构**

程序开始为函数与变量的声明区 程序常声明的一些函数:

```
prototype ShowDialogs();
//显示安装向导对话框

prototype MoveFileData();
//移动文件数据

prototype HandleMoveDataError( NUMBER );
//移动数据出错处理

prototype ProcessBeforeDataMove();
//移动文件数据前的处理

prototype ProcessAfterDataMove();
//移动文件数据后的处理

prototype SetupRegistry();
//安装注册,用户可在此加入一些代码,通常用于对注册表的操作

prototype SetupFolders();
//安装生成快捷方式,通常用户可在此加入生成快捷方式的代码

prototype CleanUpInstall();
//安装完成后清除临时文件

prototype SetupInstall();
//安装的实际过程

prototype SetupScreen();
// 设置安装过程的屏幕显示(包括背景颜色,字体等)
```

```
prototype CheckRequirements();
// 检查安装需求(包括硬盘空间,操作系统平台等)

prototype DialogShowSdWelcome();
//显示"欢迎"对话框窗口

prototype DialogShowSdLicense();
//显示许可信息的对话框

prototype DialogShowSdRegisterUserEx();
//显示用户安装注册的对话框

prototype DialogShowSdAskDestPath();
//显示"安装路径选择"对话框

prototype DialogShowSdSetupType();
//显示"安装类型选择"对话框

prototype DialogShowSdComponentDialog2();
// 当用户选择"定制"安装时,用于显示供用户选择的组件对话框

prototype DialogShowSdSelectFolder();
//显示快捷方式文件夹选择的对话框

prototype DialogShowSdFinishReboot();
//显示"安装完成重启动"对话框
```

C. 流程控制

流程控制语句:

1. abort (终止 SETUP 程序)
2. exit (退出 SETUP 程序)
3. for. . endfor (循环里面的语句,直到指定次数为止)

事例:

```
for iCount = 1 to 10
  MessageBox ("You will see this 10 times", INFORMATION);
endfor;
for j = 20 downto 10 step 5
  MessageBox ("You will see this three times", INFORMATION);
endfor;
```